

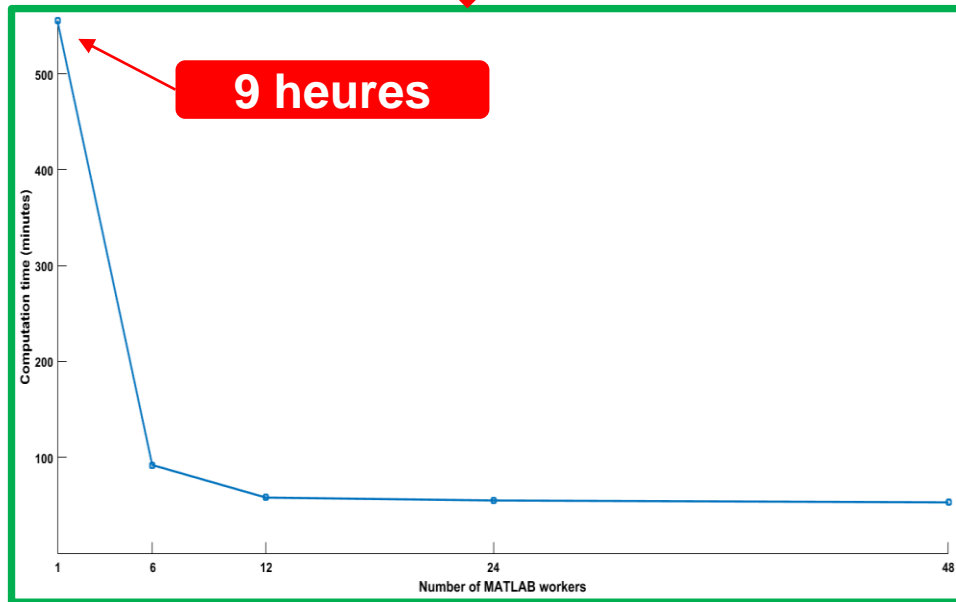
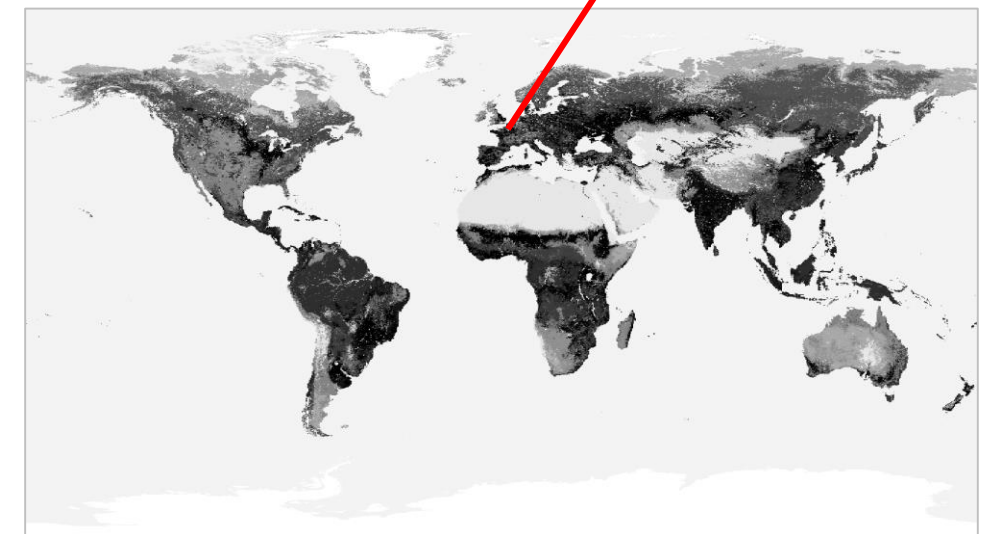
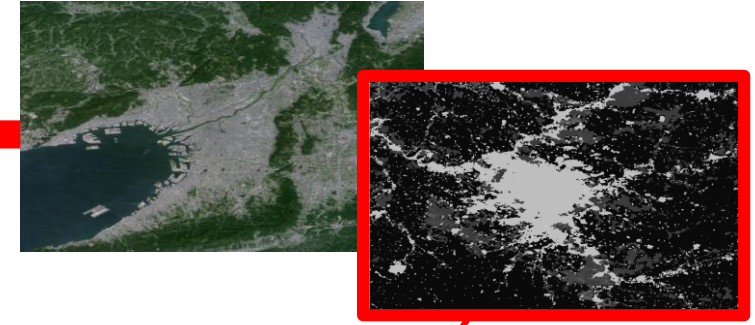
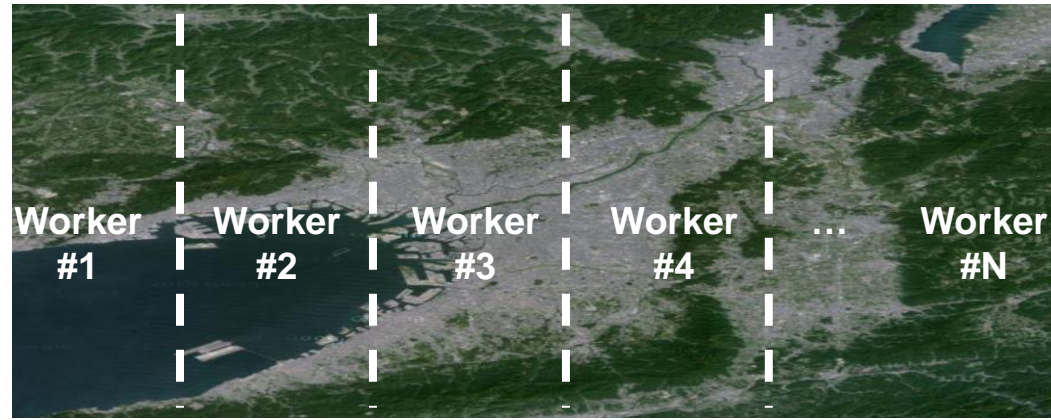
Parallel Computing & Big Data with MATLAB

Lyamine Hedjazi
Nadia Bedjaoui

Application Engineer
Academic Engineer

Drastic Time Analysis Reduction of Urban Areas

Parallel Image Processing Approach



9 heures

2 minutes

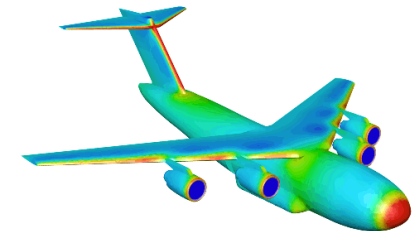
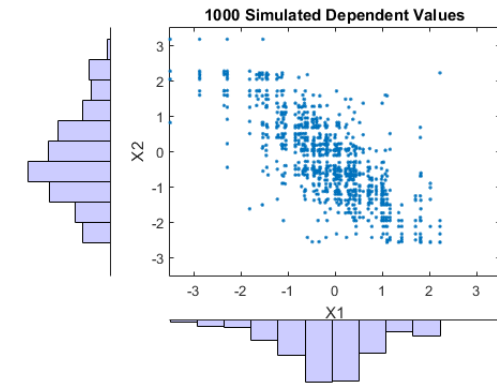
Parallel Computing can help you to...

Increase Productivity

- **Reduce computation times**
 - Accelerate your simulations
 - By taking advantage of unused CPU cores and/or GPUs

Increase Accuracy

- **Run more simulations** within the same timeframe
 - Test more values for parameter calibration and design optimization
 - 2x more runs in Monte Carlo simulations
 - ⇒ 41% increase in accuracy
- **Run larger simulations** within the same timeframe
 - Use finer meshes for Finite Element Analysis
 - Not enough memory on a single computer?
 - ⇒ Distribute the mesh on a cluster!



How much faster can my computations run?

- For an application that is well-suited for parallel computing...
 - Monte Carlo simulations
 - Parameter calibration / Design optimization
 - Linear algebra on very large matrices
 - Processing of large images and/or large collection of images
- ... the speed-up mainly depends on the **number of available CPU cores**



Laptop
1.5x to 3x



Workstation
1.5x to 20x



HPC cluster (multiple computers)
10x to 1000x or even more!

MathWorks Solutions for Parallel Computing

Parallel Computing Toolbox

- MATLAB Toolbox
- Allows to write parallel applications by giving access to parallel programming constructs
- Take advantage of **local** CPU cores and GPUs



Laptop
1.5x to 3x



Workstation
1.5x to 20x

MATLAB Distributed Computing Server

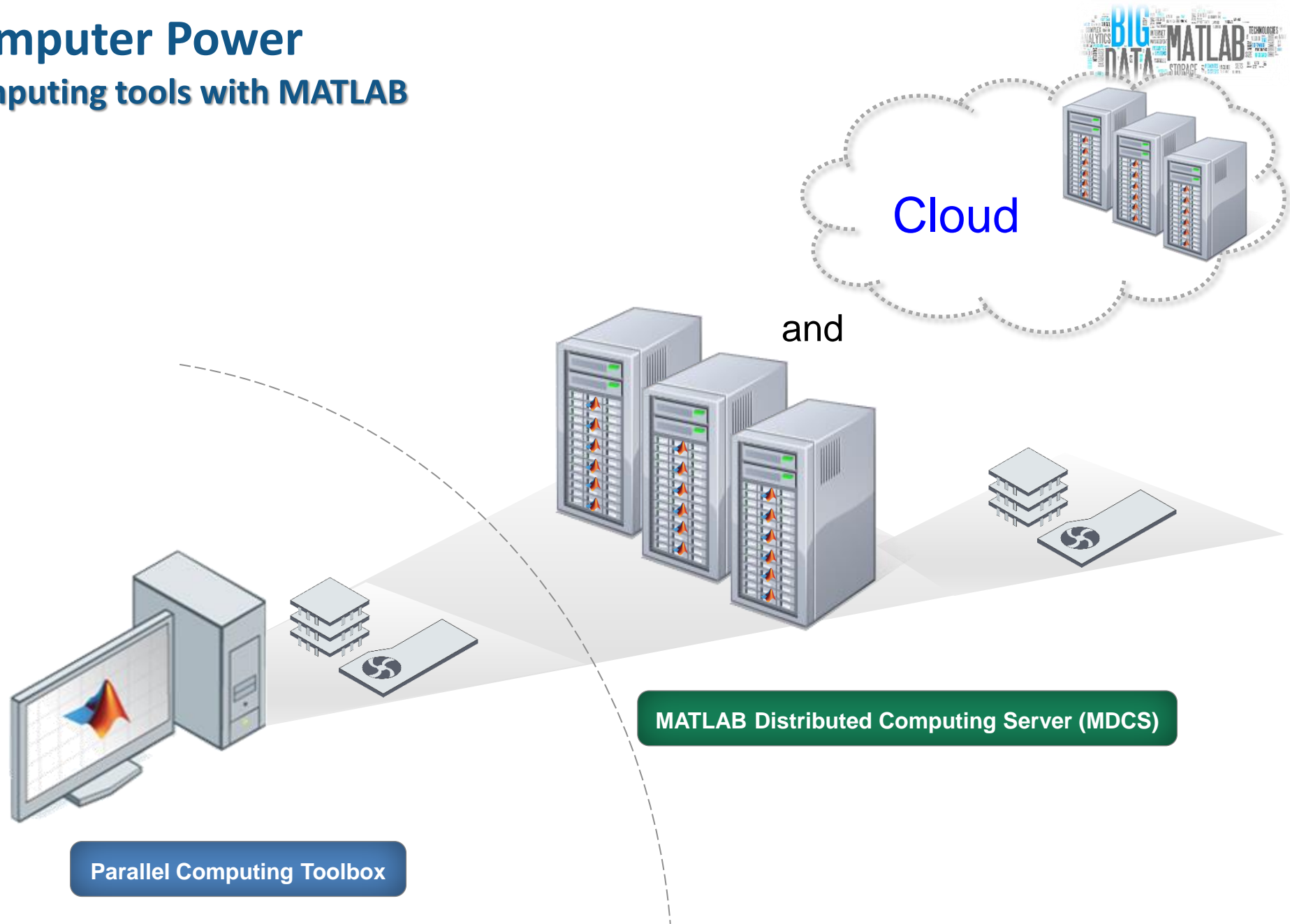
- Standalone product
- Run computations on remote resources
- With MDCS, a single application can be executed on multiple computers



HPC cluster (multiple computers)
10x to 1000x or even more!

Scale Computer Power

Parallel Computing tools with MATLAB



Scaling-up locally

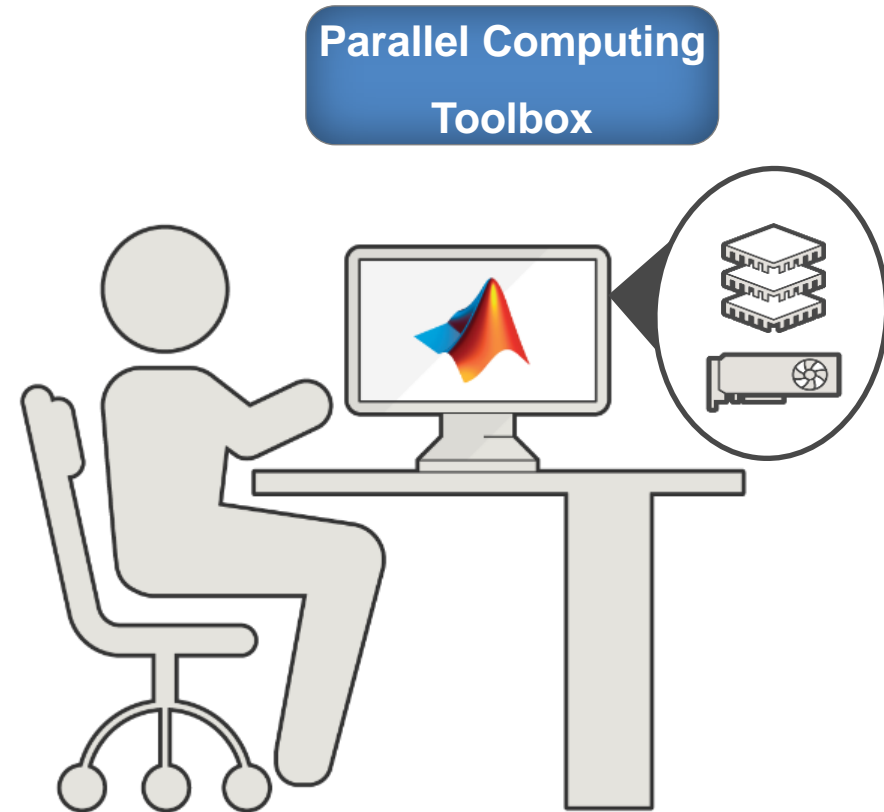
Use the full processing power of desktops (CPUs, GPUs)

- Leverage GPU accelerator

```
d=gpuDevice()
```

- Leverage Multicore CPU

```
p=parpool('local');
```



Programming Parallel Applications

Leverage GPU Accelerators

Ease of Use

- Built-in GPU support
 - Neural Network Toolbox
 - Many signal & image processing features
- GPU-accelerated functions
 - Transfer data to the GPU and run computations on the GPU
 - **Overloaded** functions, no need to rewrite your code
 - Most linear algebra functions, FFT, statistics...
- Import CUDA code into MATLAB
 - If you have written C/C++/FORTRAN code that uses the CUDA library, you can import it in MATLAB using **MEX files**
- Important remarks
 - Only NVIDIA GPUs are supported
 - Make sure to use GPUs with full double precision support (NVIDIA Tesla)

Greater Control



Leveraging your parallel computing resources

Multicore CPUs and systems

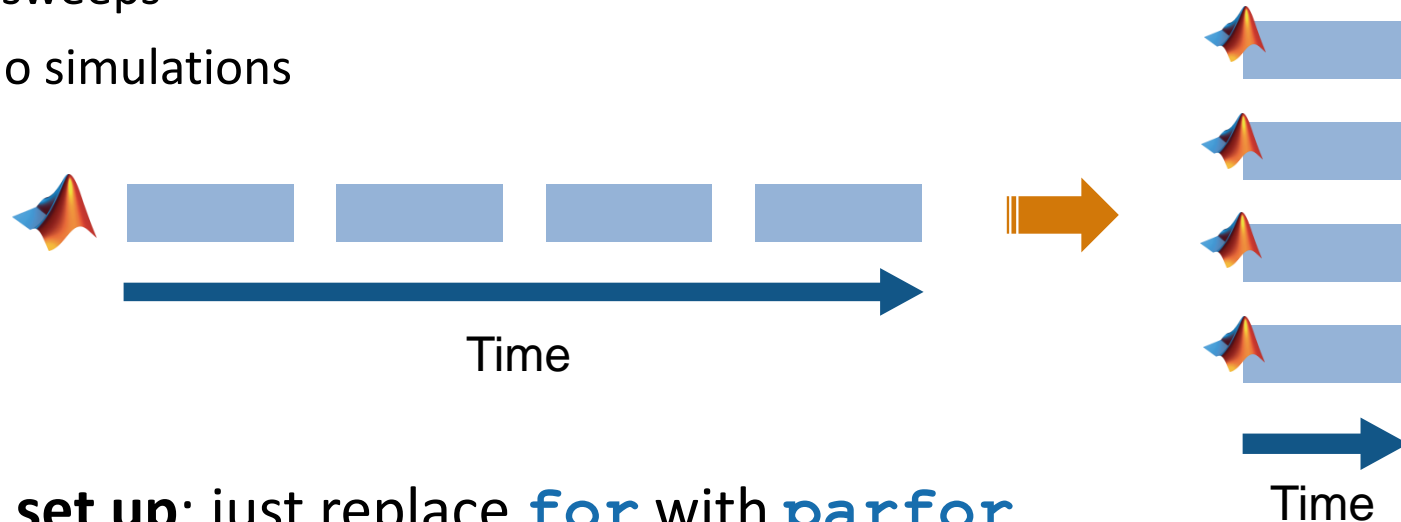


- Built-in Parallel Computing support
 - Many toolboxes can benefit from parallel speed-up
 - Optimization & Global Optimization Toolboxes
 - Statistics and Machine Learning Toolbox
 - Neural Network Toolbox
 - Image Processing Toolbox & Computer Vision System Toolbox
 - ...
 - Check the documentation for a **UseParallel** property
 - Or refer to www.mathworks.com/builtin-parallel-support
- **parfor** (parallel-for loops)
 - Speed up for loops with **independent** iterations
 - Monte Carlo simulations, parameter calibration



Parallel-for Loops

- Parallel-for loops used to **speed-up for loops** with **independent** iterations
 - Parameter sweeps
 - Monte Carlo simulations



- Very easy to set up:** just replace `for` with `parfor`
- Ideal use cases for parallel computing
 - Low overhead, no communication between tasks
 - Excellent scalability and speed-ups

Leveraging your parallel computing resources

Multicore CPUs and systems



- Built-in Parallel Computing support
 - Many toolboxes can benefit from parallel speed-up
 - Optimization & Global Optimization Toolboxes
 - Statistics and Machine Learning Toolbox
 - Neural Network Toolbox
 - Image Processing Toolbox & Computer Vision System Toolbox
 - ...
 - Check the documentation for a **UseParallel** property
 - Or refer to www.mathworks.com/builtin-parallel-support
- **parfor** (parallel-for loops)
 - Speed up for loops with **independent** iterations
 - Monte Carlo simulations, parameter calibration
- MPI-like programming constructs
 - **spmd** blocks
 - **labSend**, **labReceive**, **gop**



spmd blocks

```
spmd
    % single program across workers
end
```

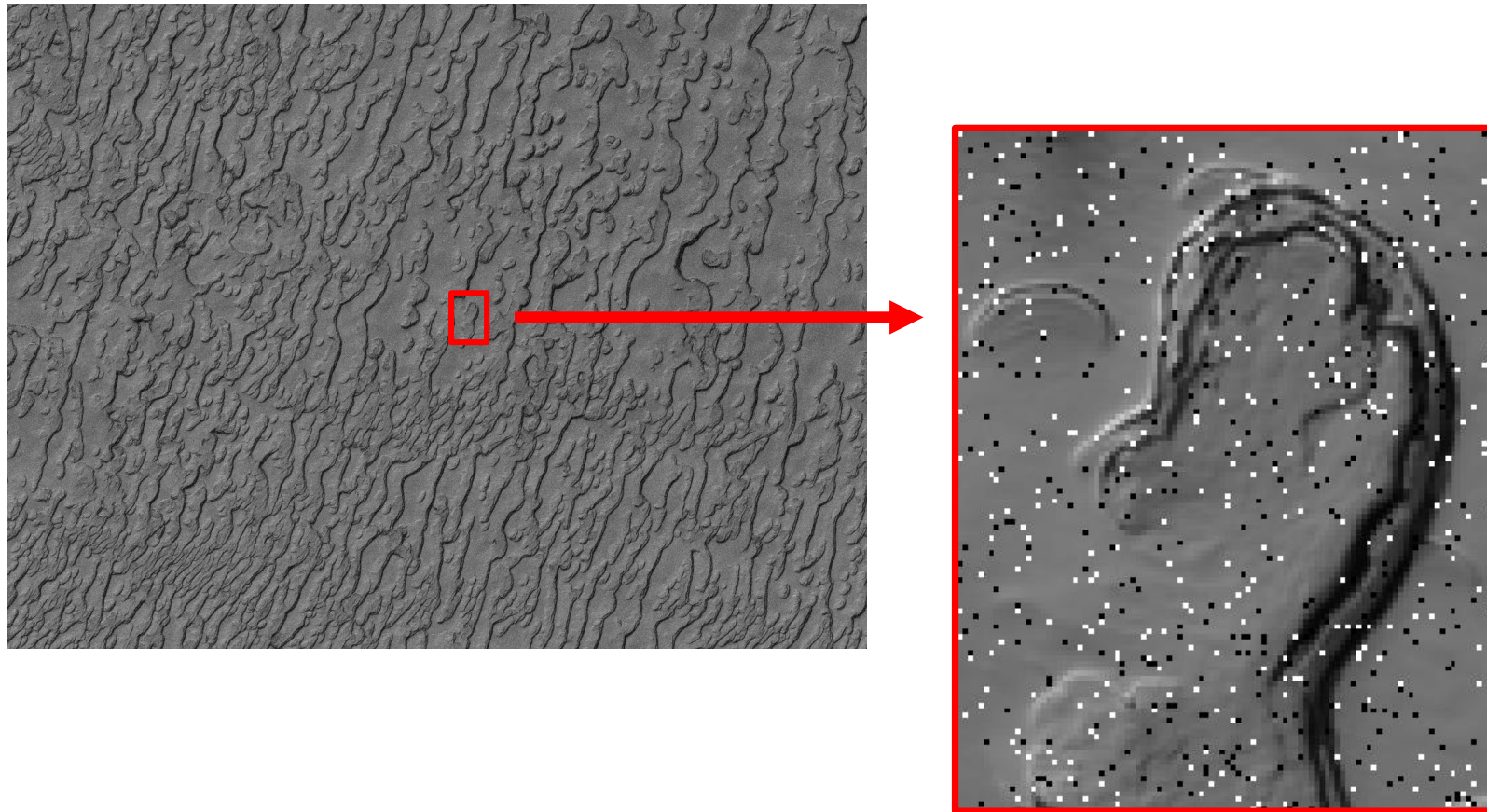
- Mix parallel and serial code in the same function
- Run on a pool of MATLAB resources
- **S**ingle **P**rogram runs simultaneously across workers
- **M**ultiple **D**ata spread across multiple workers

```
spmd
    a = labindex + numlabs;
    disp(['a = ',
        num2str(a)]);
end
```

```
>> Lab 1      a = 5
Lab 2      a = 6
Lab 3      a = 7
Lab 4      a = 8
```

Communicating tasks – Denoising a Hi-Res Image

- **Goal:** denoising a large image using median filtering



Communicating tasks – Denoising a Hi-Res Image

Median filtering

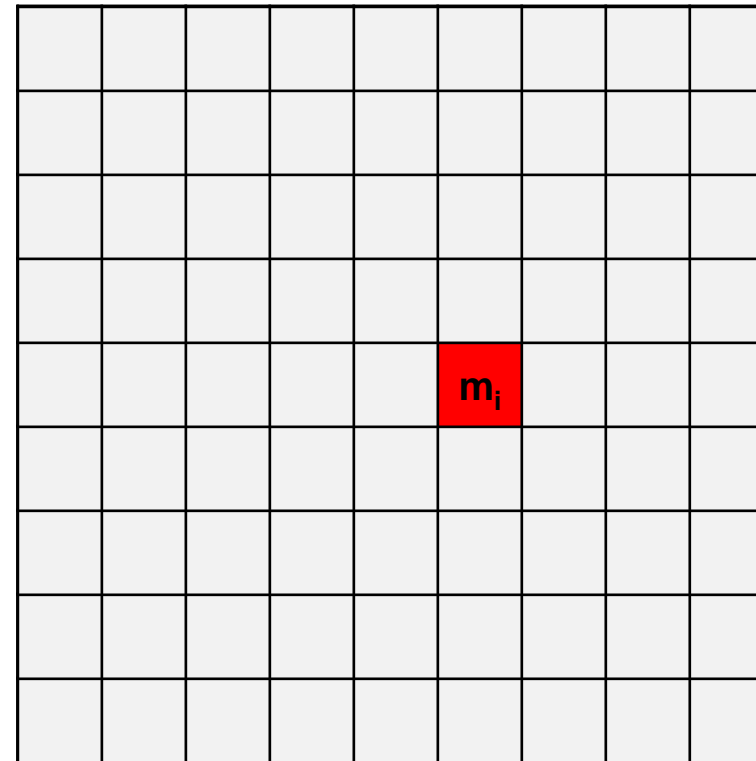
- Useful tool for removing ‘salt and pepper’ noise
- Filter is applied pixel by pixel

- **Algorithm**

- for each pixel p_i in the image
- take a square subset of width w around p_i
- compute the median value m_i of this subset
- replace the value of p_i with m_i

- **MATLAB functions**

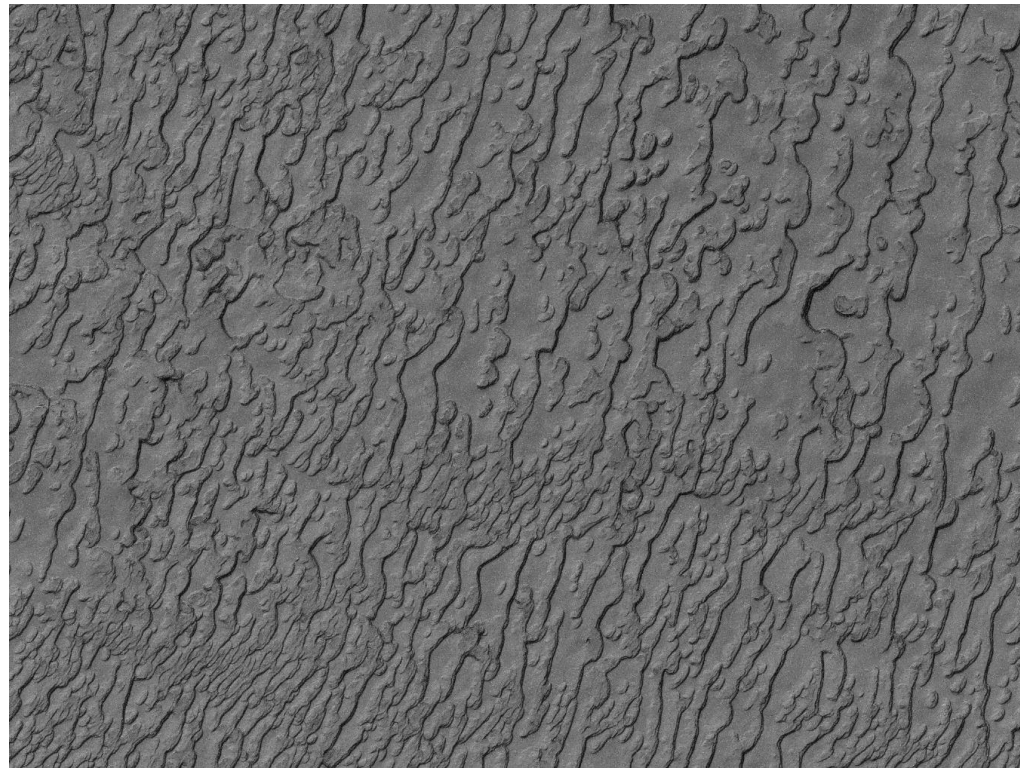
- `medfilt1` for the 1-D case
- `medfilt2` for the 2-D case



$m_i = \text{median}(\text{yellow square})$

Communicating tasks – Denoising a Hi-Res Image

Parallelization technique on multicore CPU(s)



Communicating tasks – Denoising a Hi-Res Image

Parallelization technique on multicore CPU(s)

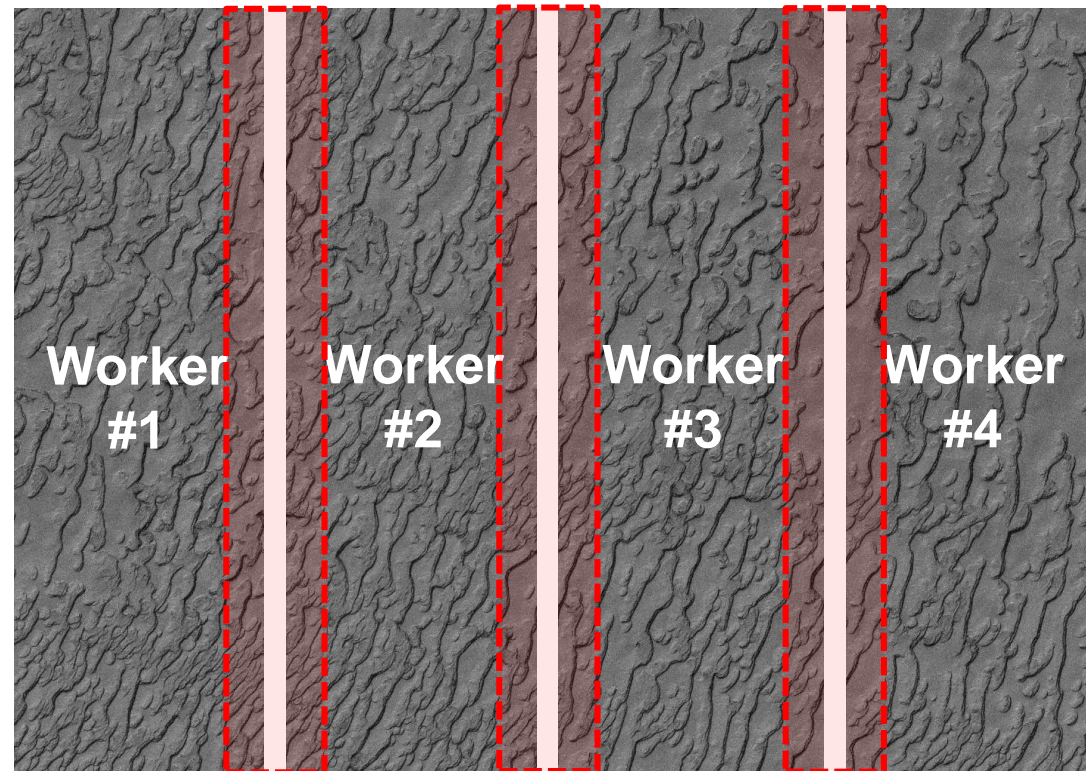
- Divide the image in multiple stripes, each processed by a different worker



Communicating tasks – Denoising a Hi-Res Image

Parallelization technique on multicore CPU(s)

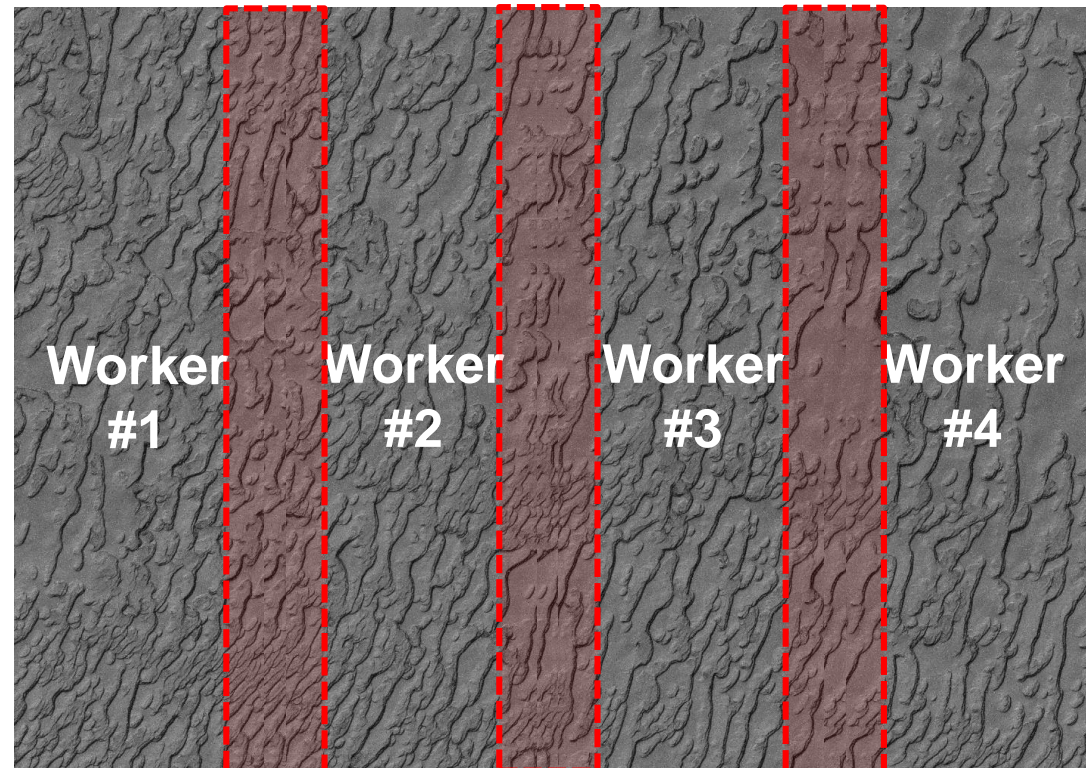
- Divide the image in multiple stripes, each processed by a different worker
- Need for overlay pixels due to the filter's algorithm



Communicating tasks – Denoising a Hi-Res Image

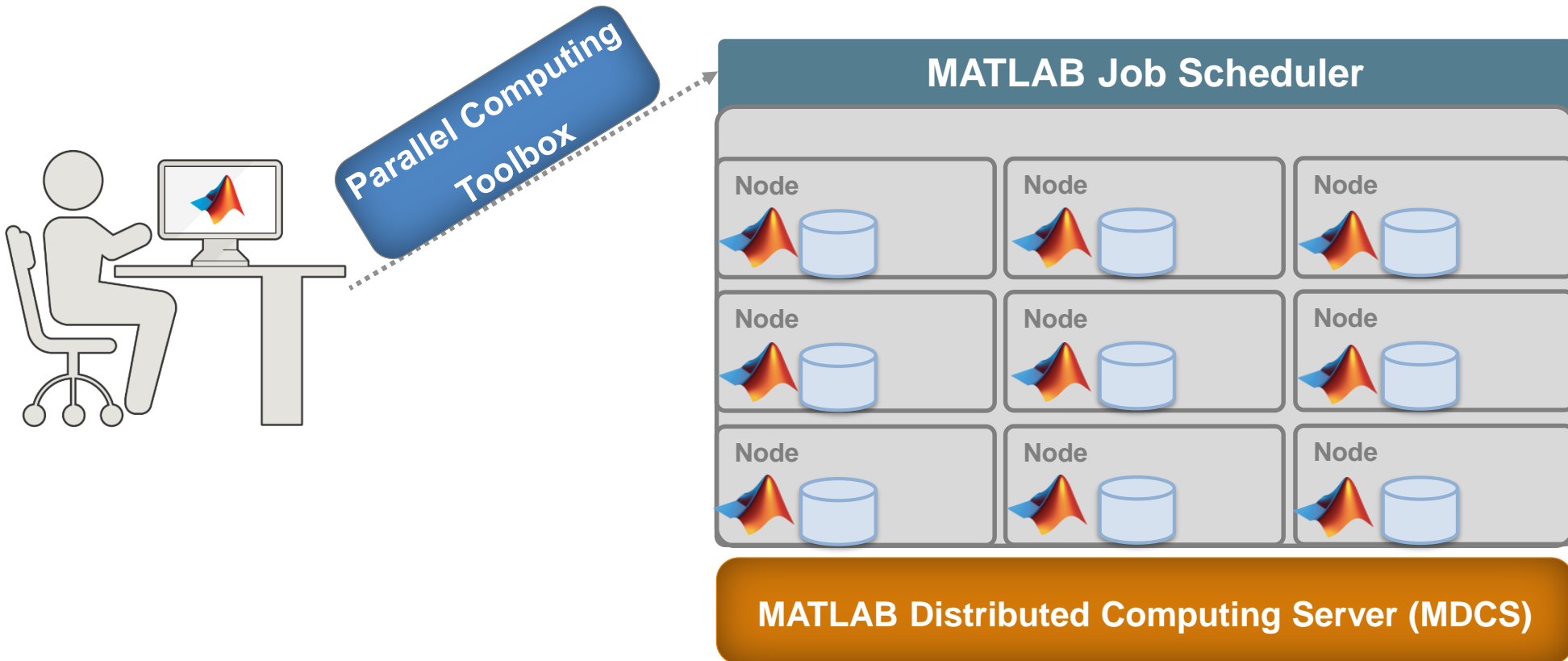
Parallelization technique on multicore CPU(s)

- Divide the image in multiple stripes, each processed by a different worker
- Need for overlay pixels due to the filter's algorithm
- Once processed, reconcatenate the stripes



Running jobs interactively on the cluster

```
cluster = parcluster('MyCluster');  
parpool(cluster);
```



Integration with third-party schedulers

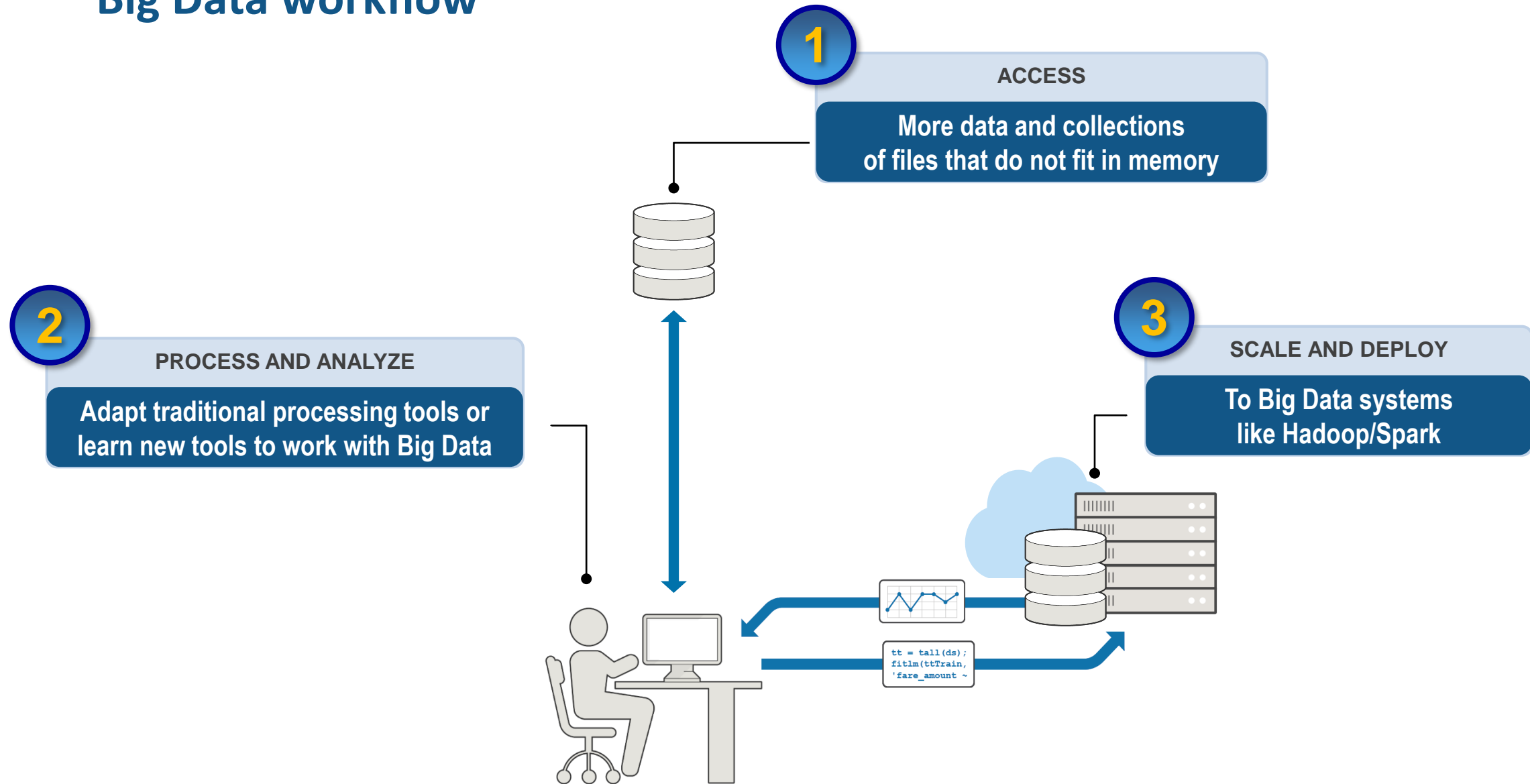
- MATLAB Distributed Computing Server includes a MATLAB job scheduler and support for other the following schedulers: SLURM*, LSF, HPC server, PBS and other generic schedulers.

- **Submission Mode with generic schedulers:**
 - **Shared** — When the client machine is able to submit directly to the cluster and there is a shared file system present between the client and the cluster machines.
 - **Remote Submission** — When there is a shared file system present between the client and the cluster machines, but the client machine is not able to submit directly to the cluster (for example, if the scheduler's client utilities are not installed).
 - **Nonshared** — When there is not a shared file system between client and cluster machines.

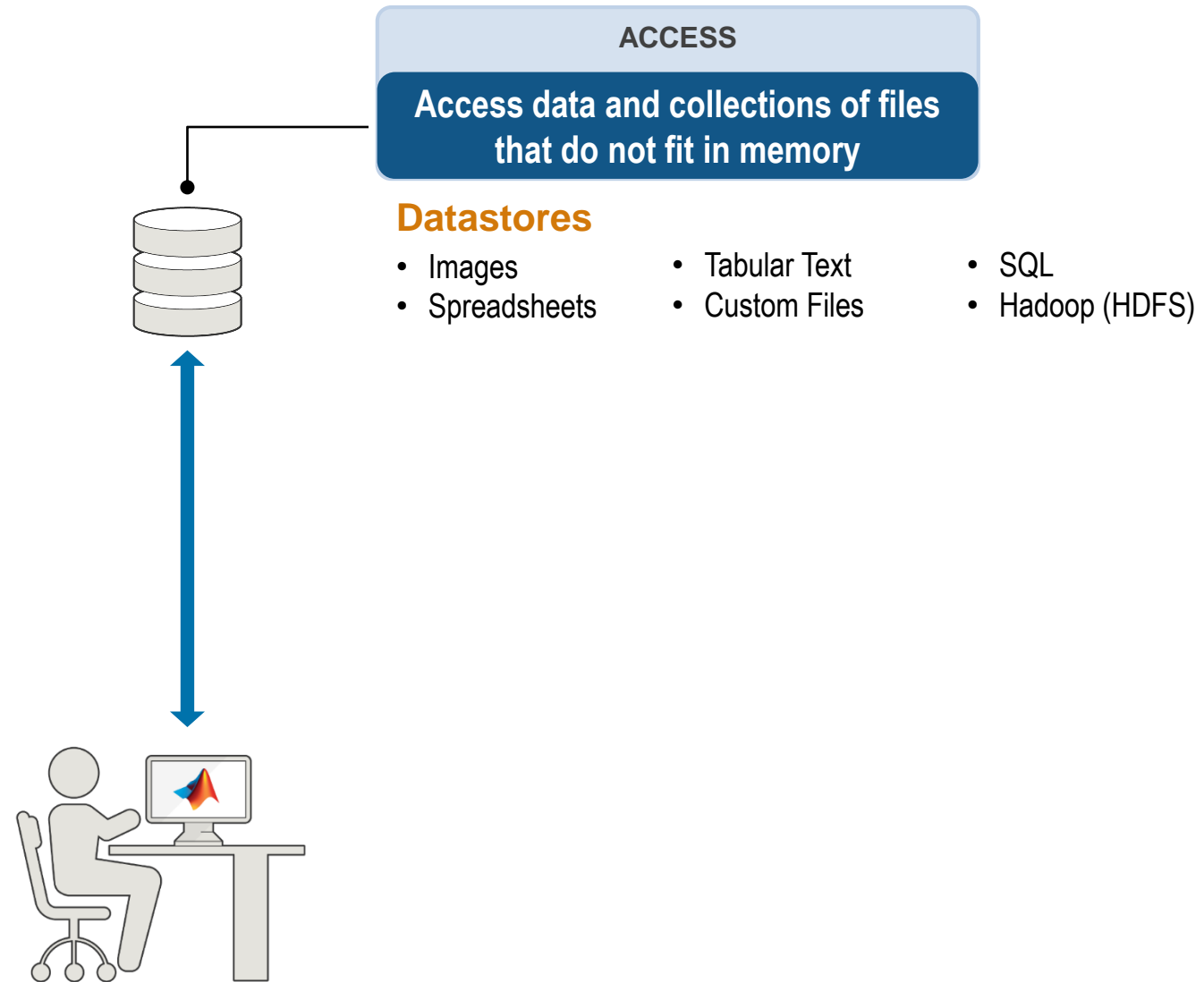
* Customizable integration uses the generic scheduler interface.

What about scalability to an increasing volume of data (Big Data)?

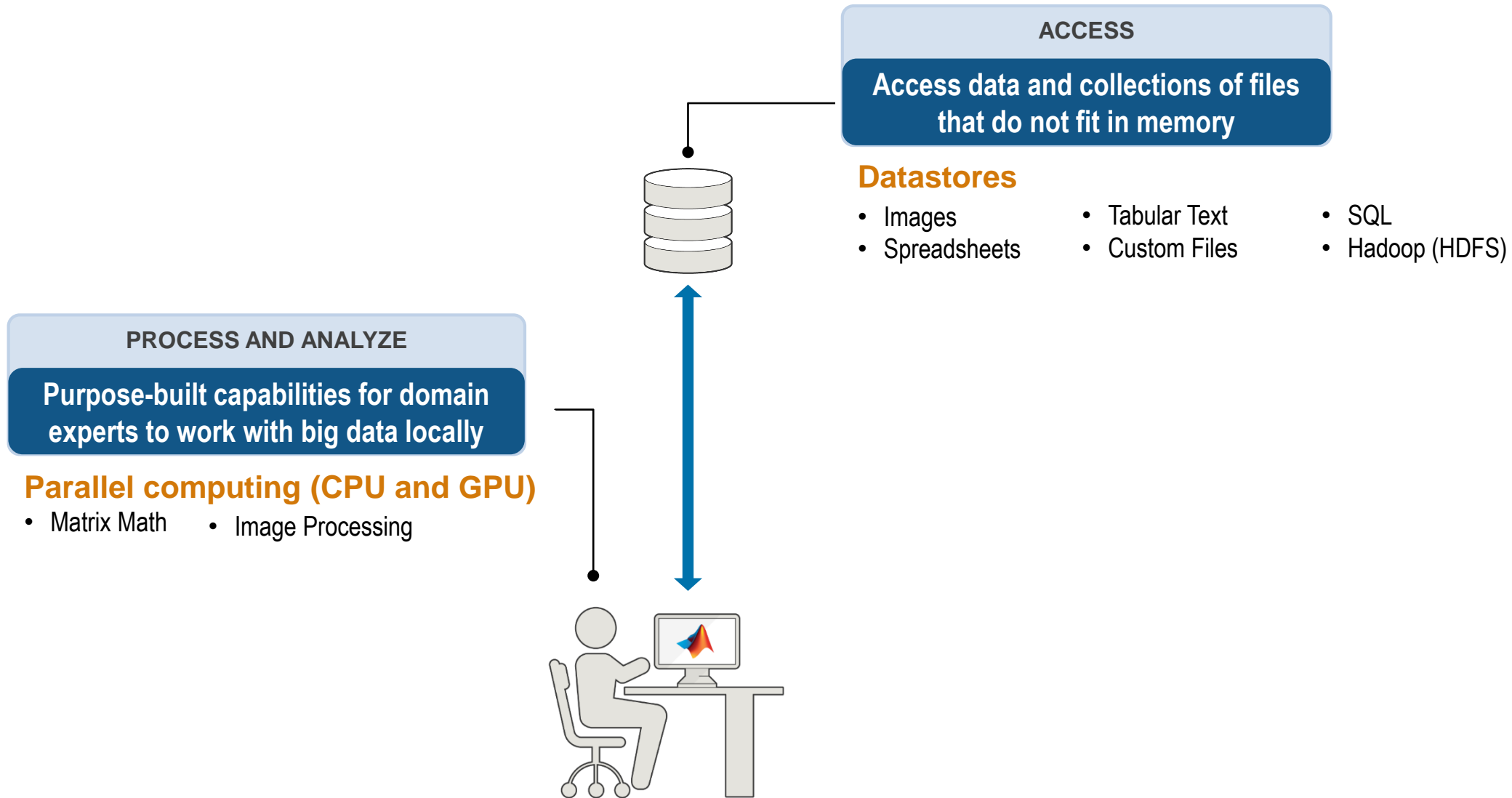
Big Data workflow



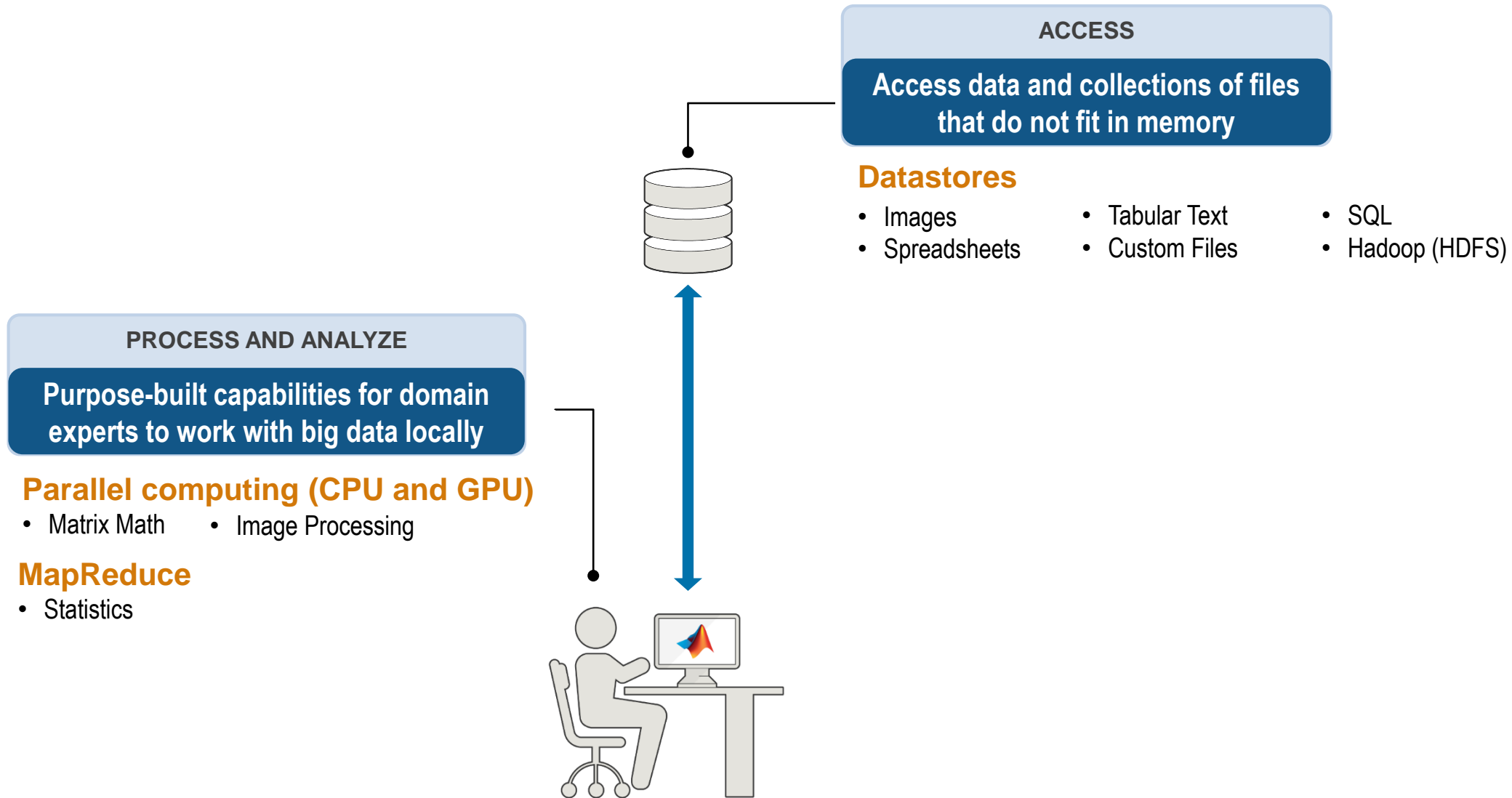
Big Data capabilities in MATLAB



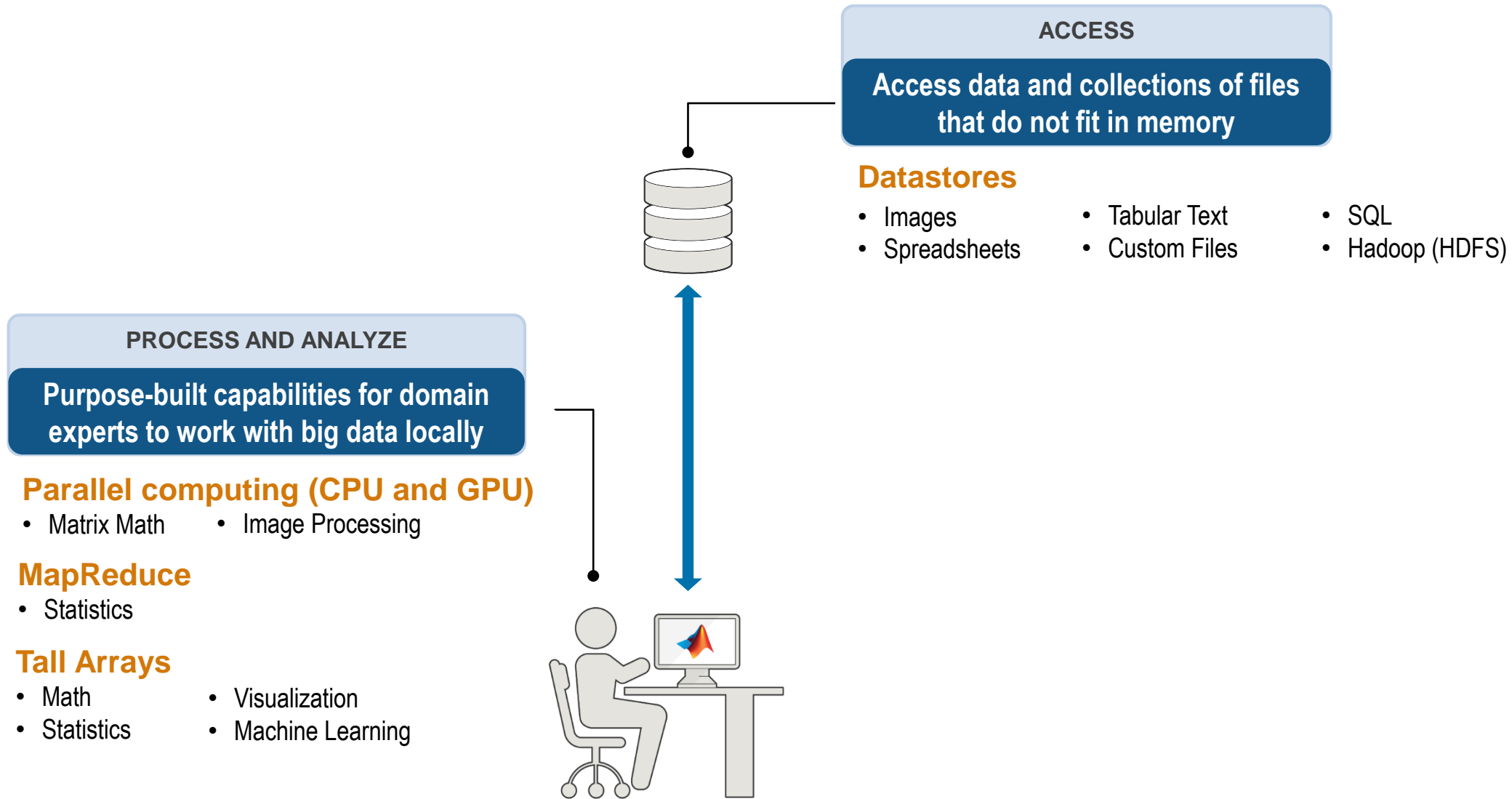
Big Data capabilities in MATLAB



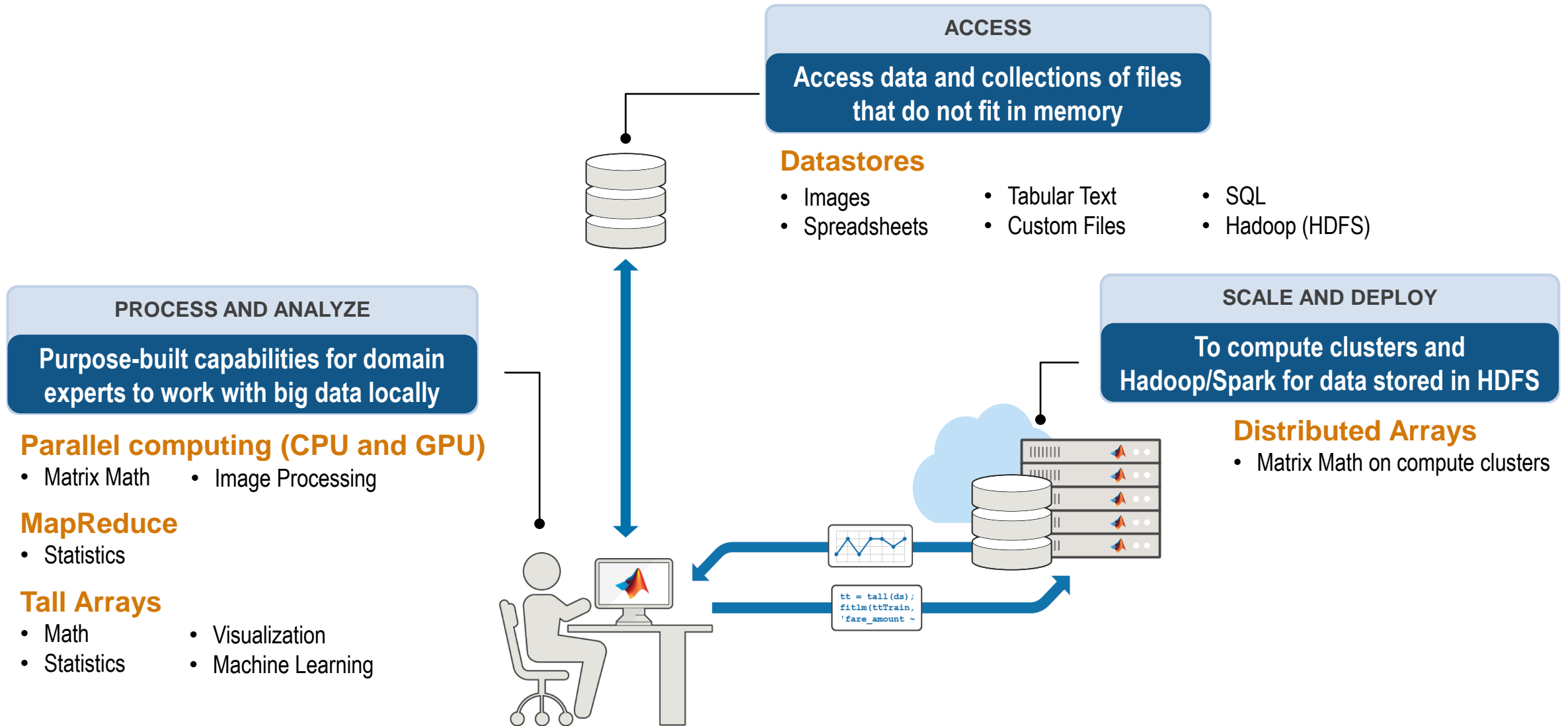
Big Data capabilities in MATLAB



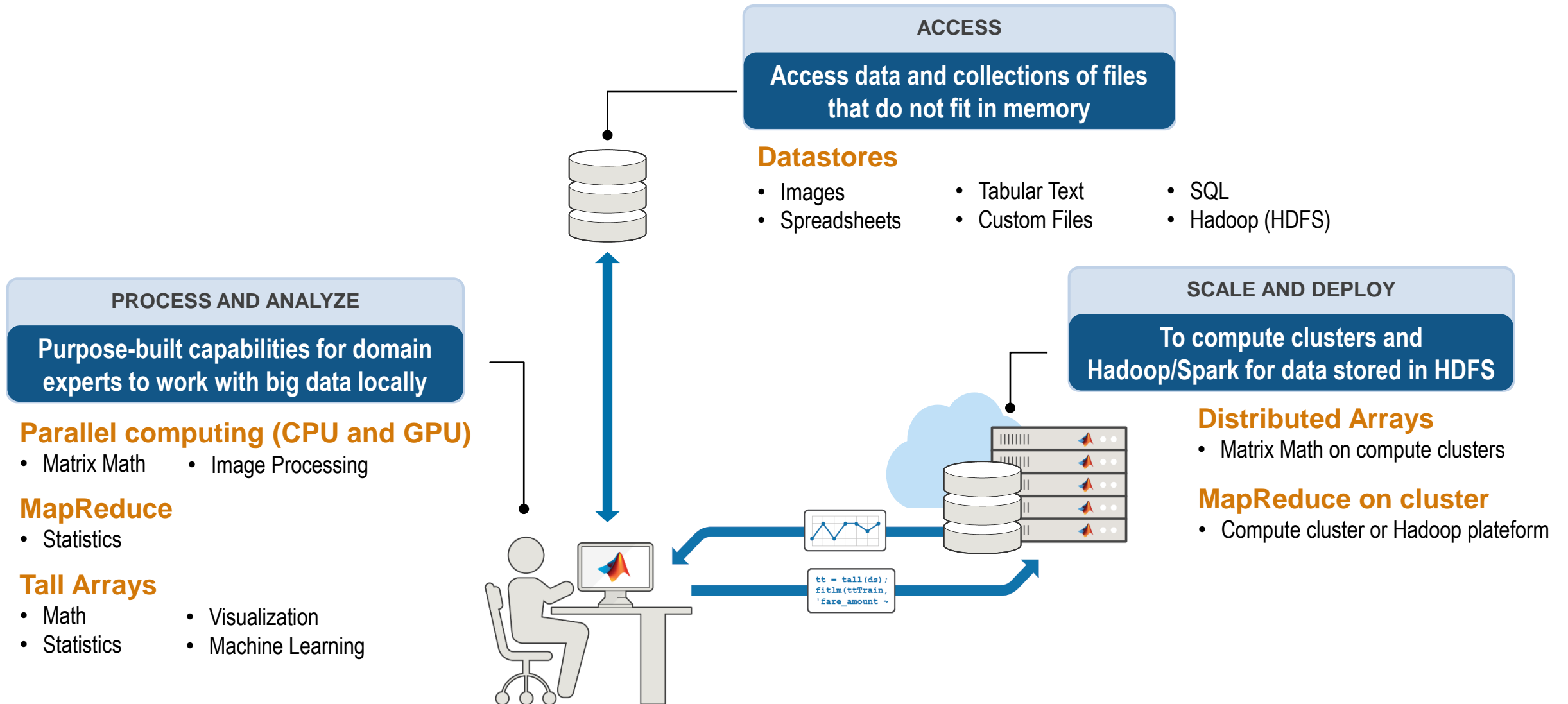
Big Data capabilities in MATLAB



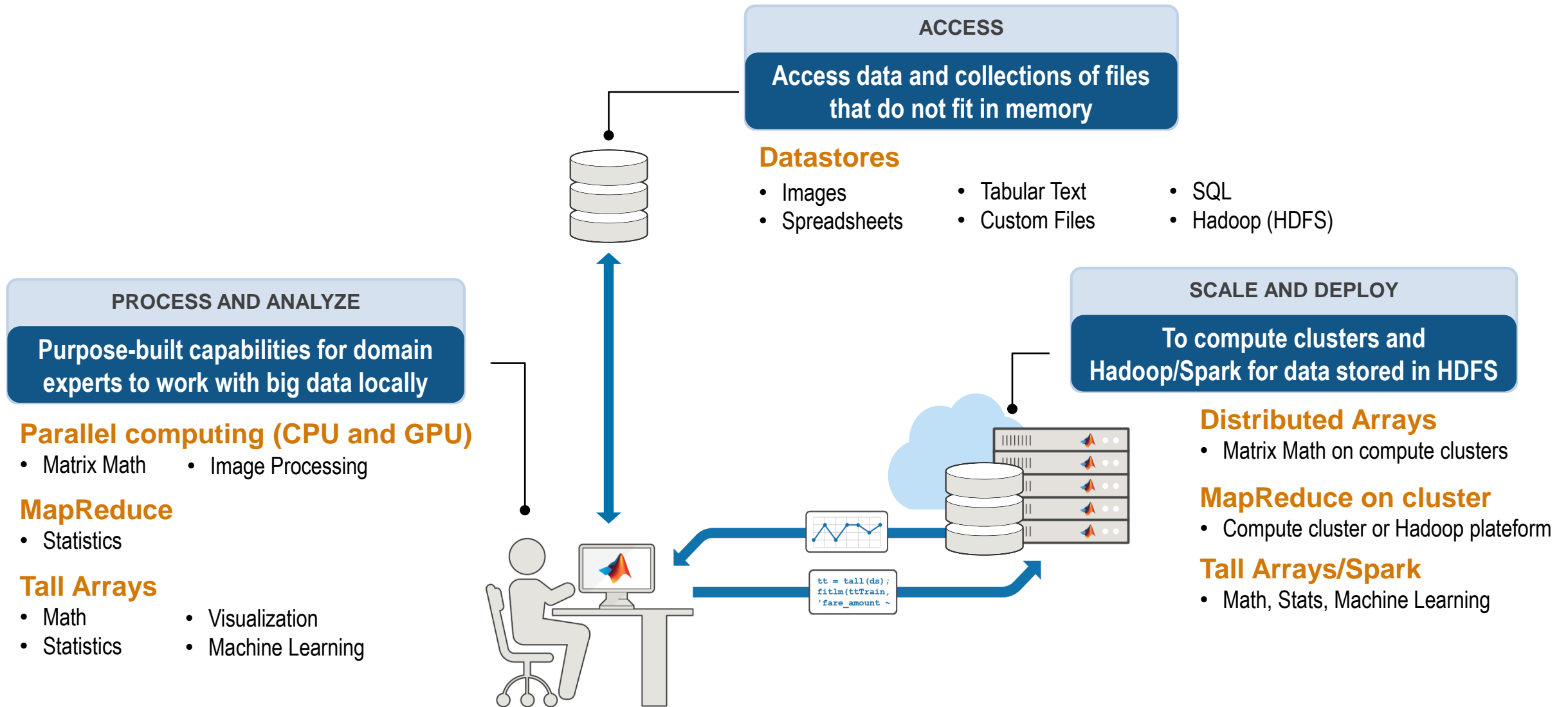
Big Data capabilities in MATLAB



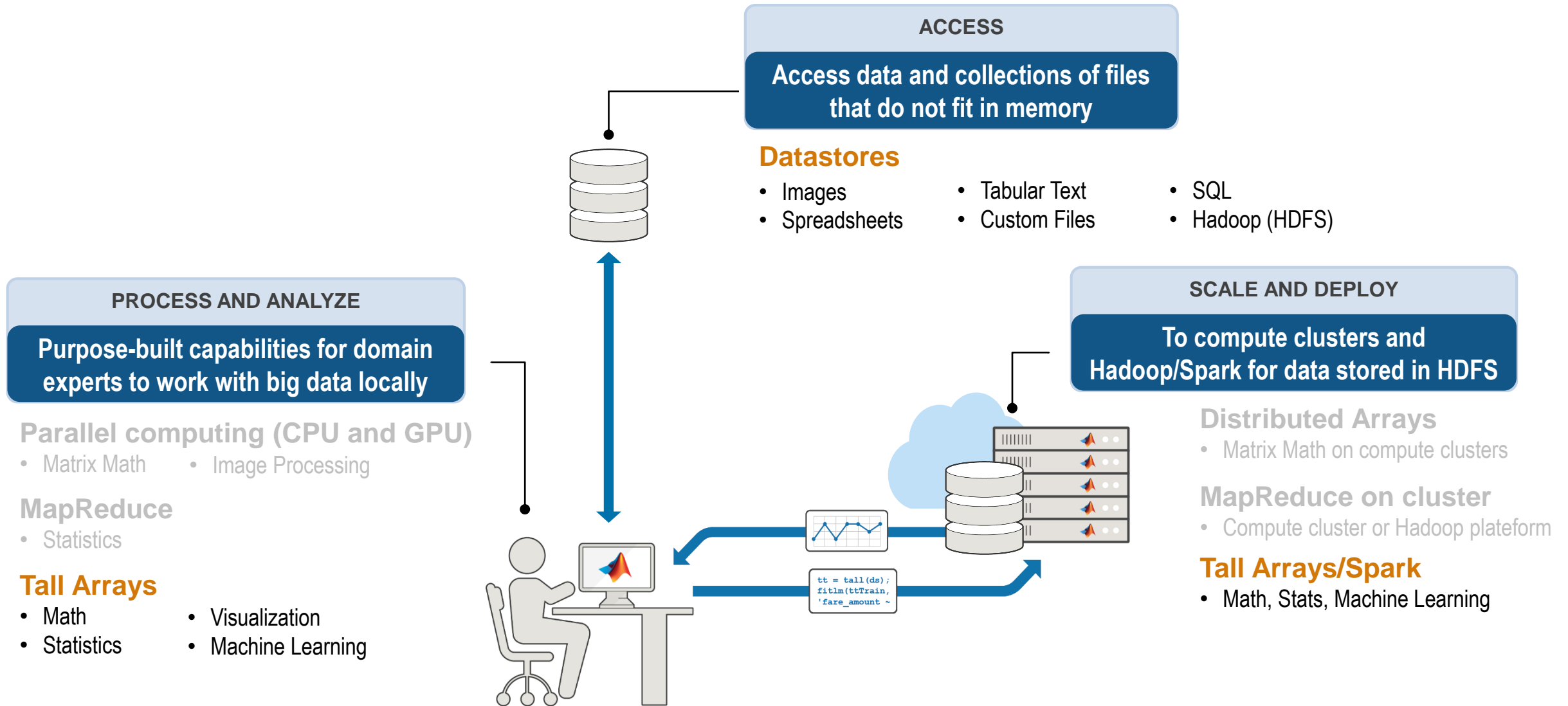
Big Data capabilities in MATLAB



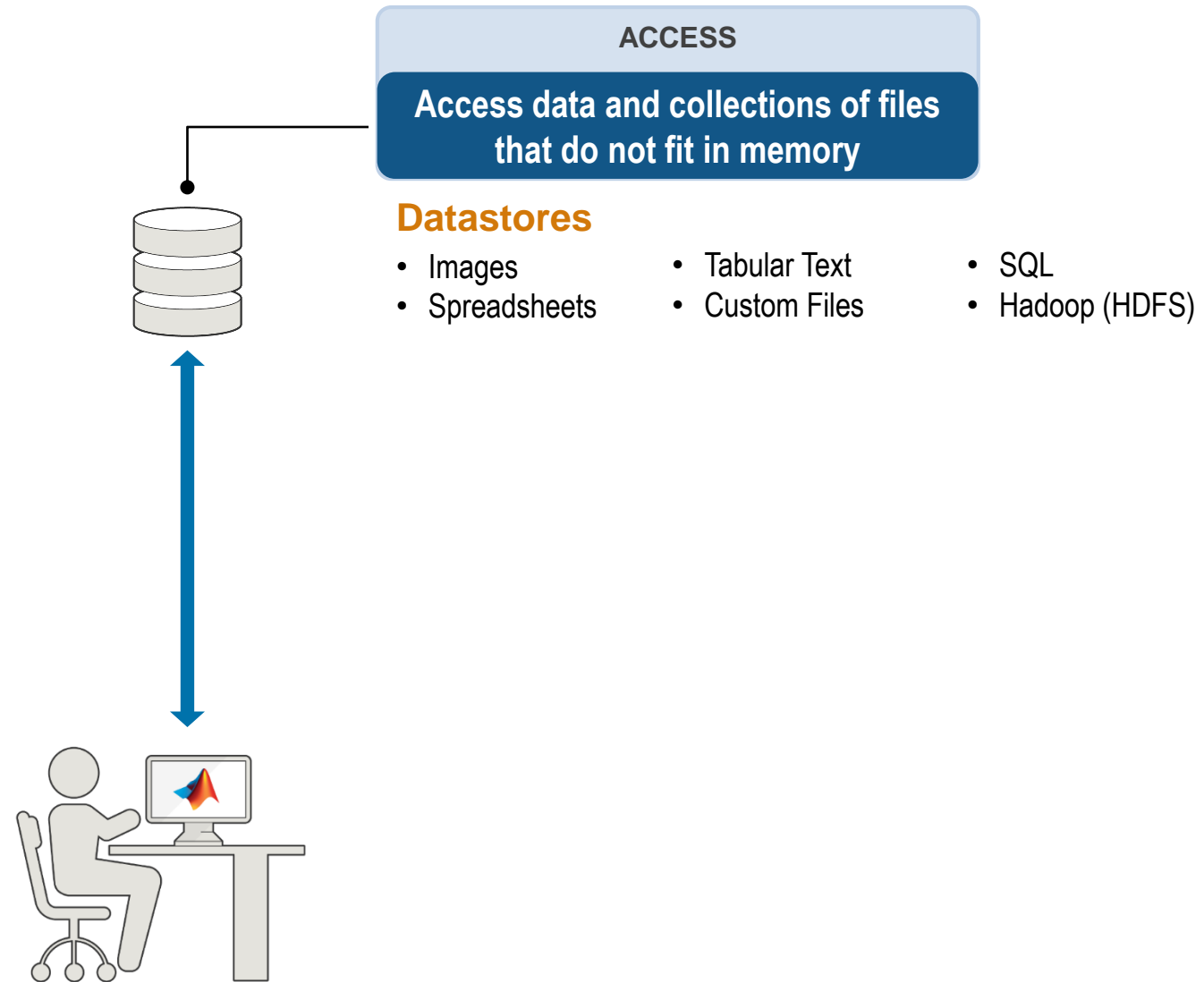
Big Data capabilities in MATLAB



Big Data capabilities in MATLAB: Tall Arrays Workflow

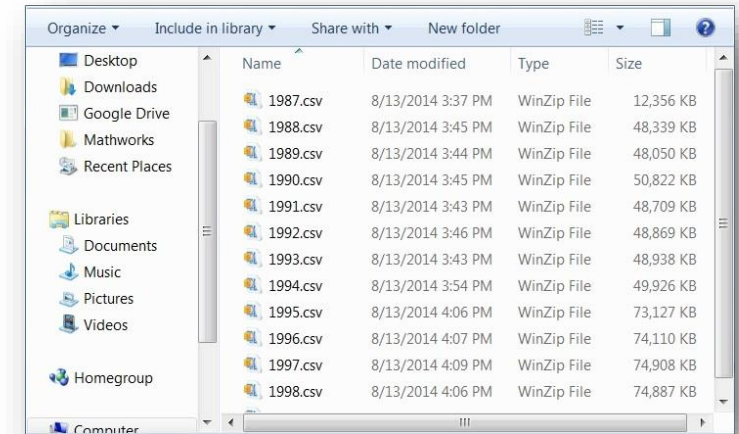


Big Data capabilities in MATLAB



Access Big Data datastore

- Easily specify data set
 - Single text file (or collection of text files)
 - Hadoop HDFS
 - Images
 - Database (SQL, No SQL,)
- Preview data structure and format
- Select data to import using column names
- Incrementally read subsets of the data



```
>> preview(ds)
ans =
```

Year	Month	DayofMonth	DayOfWeek
1987	10	21	3
1987	10	26	1
1987	10	23	5
1987	10	23	5

```
airdata = datastore('*.*.csv');
airdata.SelectedVariables = {'Distance', 'ArrDelay'};

data = read(airdata);
```

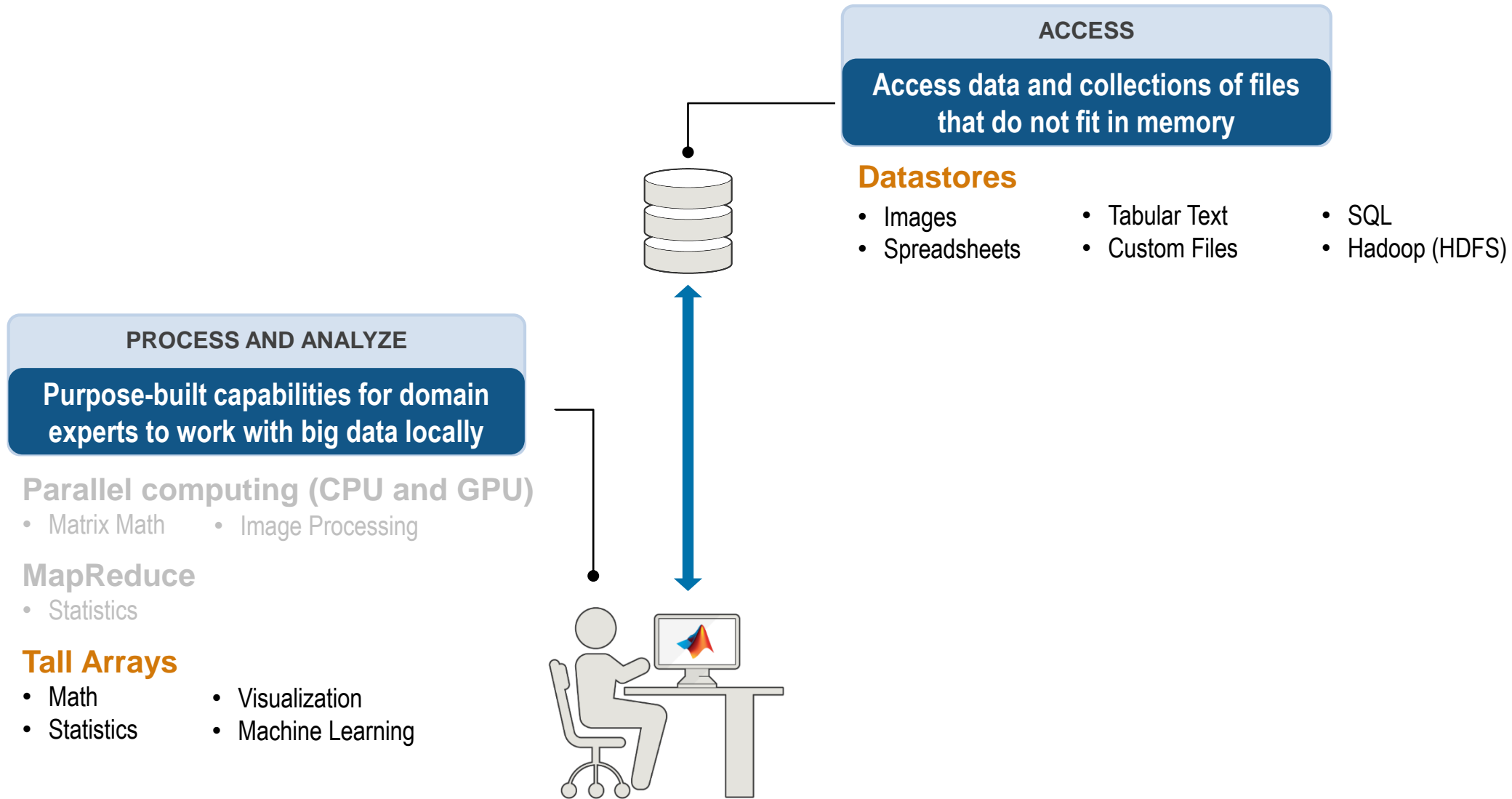
Example: Access Big Data with MATLAB

VendorID,	tpep_pickup_datetime,	tpep_dropoff_datetime,	passenger_count,	trip_distance,	pickup_longitude,	picku
2,	2015-01-07 07:40:20,	2015-01-07 08:04:45,	6,	9.12,	-73.9524536132812,	40.78
2,	2015-01-21 22:49:50,	2015-01-21 23:17:11,	6,	5.63,	-74.0083694458008,	40.73
1,	2015-01-05 23:04:30,	2015-01-05 23:15:00,	1,	2.9,	-73.8632125854492,	40.76
1,	2015-01-11 22:20:43,	2015-01-11 22:23:02,	1,	0.8,	-73.9577560424805,	40.76
2,	2015-01-24 00:34:59,	2015-01-24 00:38:39,	1,	0.65,	-73.9916687011719,	40.73
1,	2015-01-25 19:09:57,	2015-01-25 19:18:02,	1,	1.5,	-73.9983825683594,	40.72
1,	2015-01-02 23:24:13,	2015-01-02 23:27:30,	1,	1,	-73.9963912963867,	40.75
2,	2015-01-21 06:46:23,	2015-01-21 06:47:56,	1,	0.63,	-73.9913635253906,	40.77
2,	2015-01-23 19:32:33,	2015-01-23 19:49:56,	3,	2.52,	-73.999382019043,	40.73

- **Objective:** Access data and calculate the average trip duration of New York taxi.
- **Description:**
 - Monthly taxi ride log files
 - The full data set is **big** (~11 GB)
- **Approach:**
 - Preprocess data
 - Calculate average trip duration

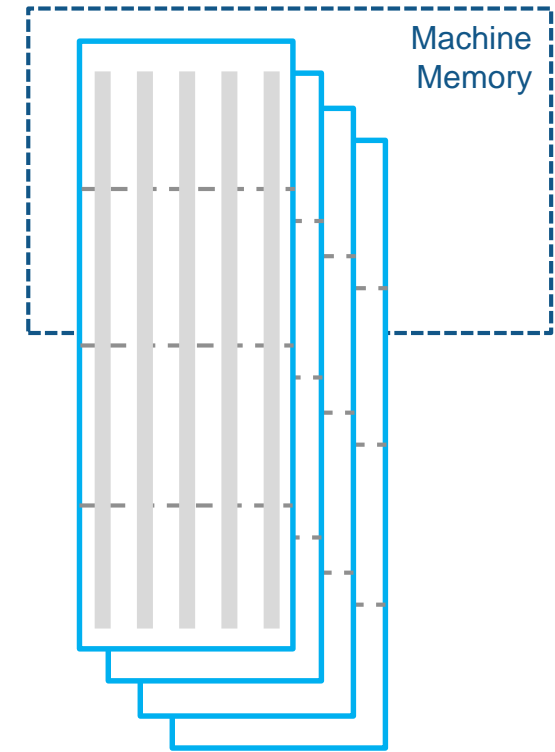


Big Data capabilities in MATLAB: Integration with SPARK



Analyze Big Data using Tall Arrays

- New data type in MATLAB
- Applicable when:
 - Data is **columnar** – with **many** rows
 - Overall data size is **too big to fit into memory**
 - Operations are mathematical/statistical in nature
- Statistical and machine learning applications
 - Hundreds of functions supported in MATLAB and Statistics and Machine Learning Toolbox
- Prototype your algorithms on desktop to easily scale your big data applications to compute clusters and clusters with Hadoop/Spark



Tall Data

What actually happens with tall arrays in MATLAB?

- Datastore

- Works as it always has

```
>> fileLoc = 'alltaxiData\*.csv';
>> ds = datastore(fileLoc);
```

- Tall array

- Pointer to the data; not actually available in memory

```
>> tt = tall(ds);
```



tall

- Functions

- Operations are queued up; execution is deferred

```
>> tt.mins = minutes(tt.dropoff-tt.pickup);
```

- Gather step

- Function that forces execution and gathers results back to the workspace (*result must fit in memory*)
- Happens with **gather** function, visualizations, and fitting/modeling functions

```
>> mn = gather(tt.mins)
```

Local disk
Shared folders
Databases



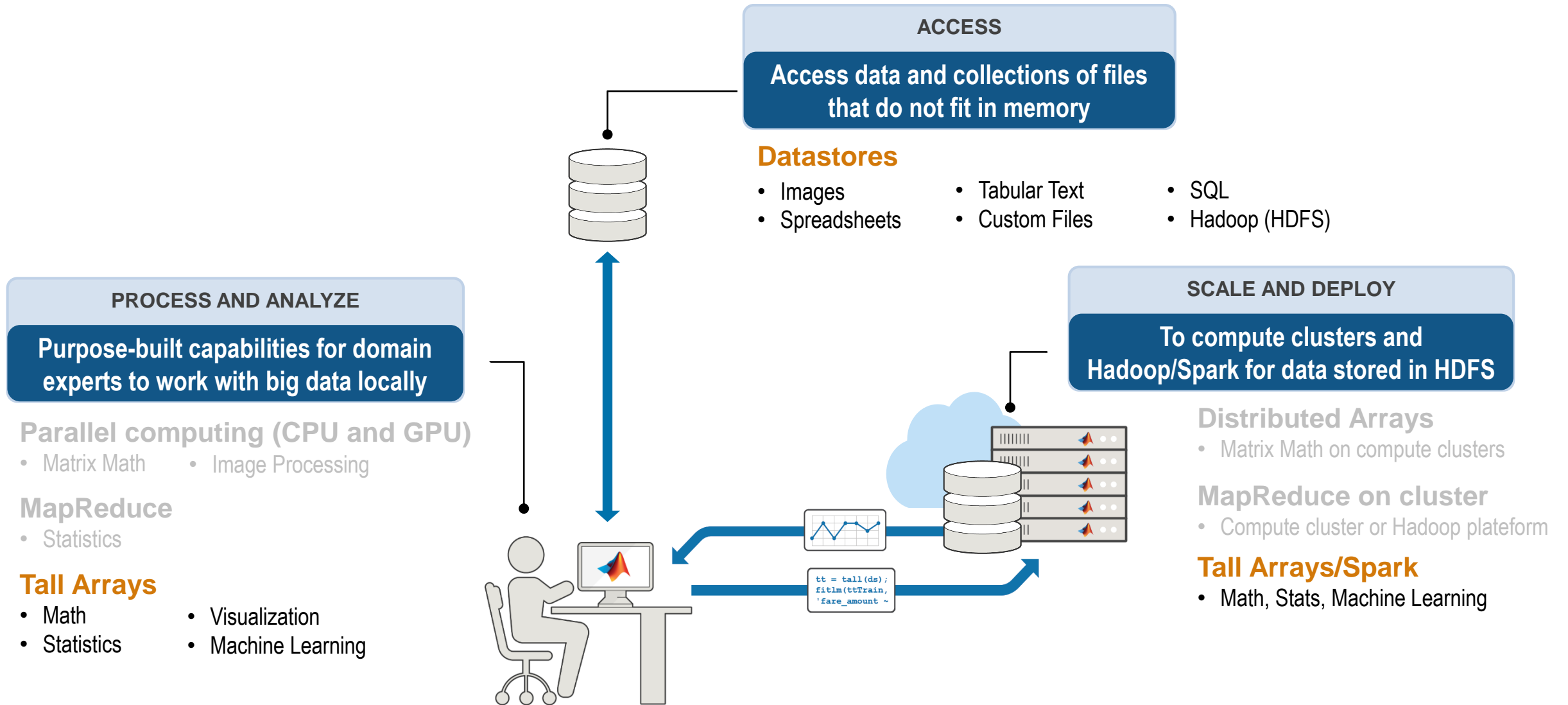
Spark + Hadoop

Example: Tall Arrays

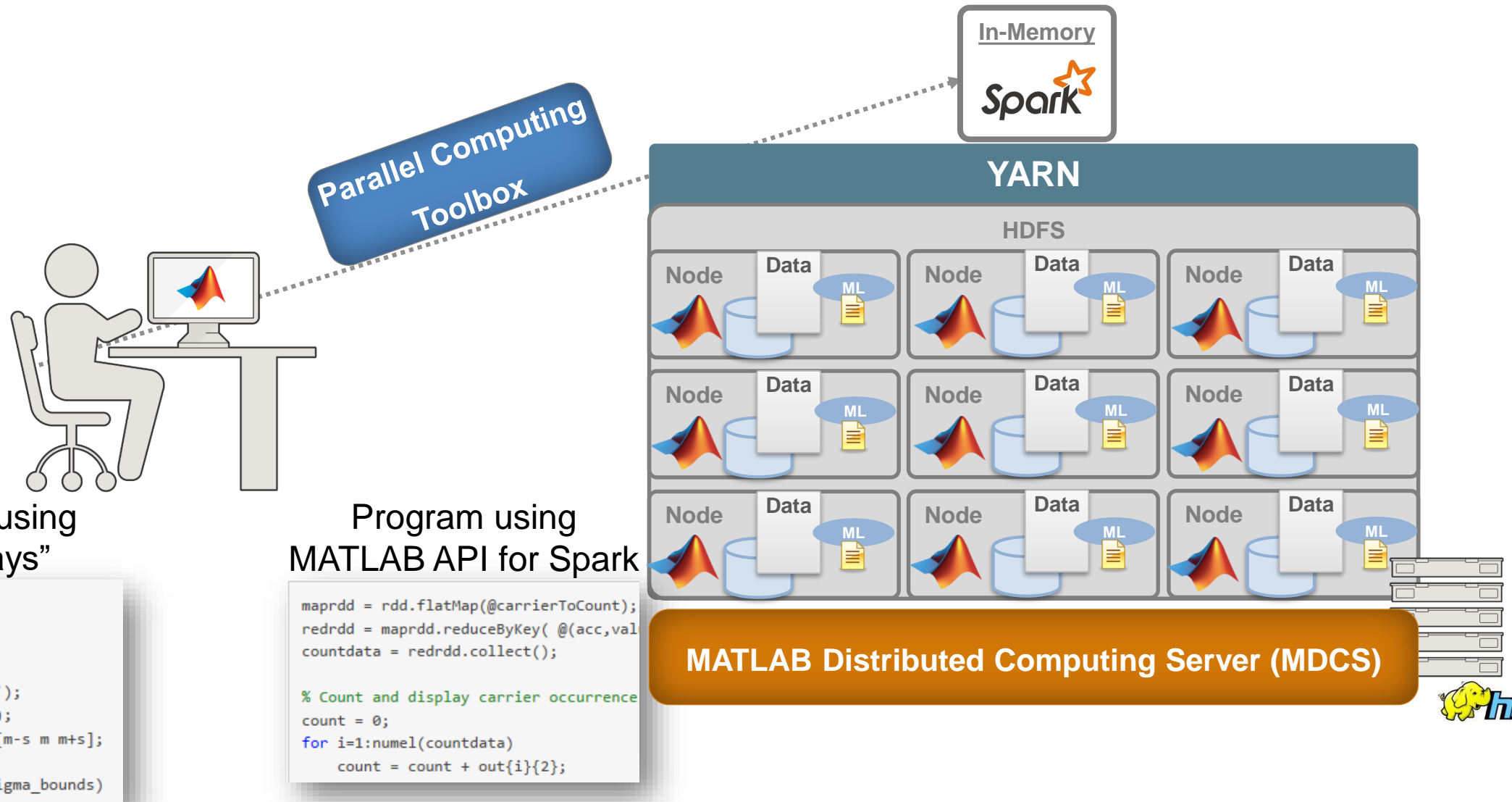
- **Objective:** Create a model to predict the cost of a taxi ride in New York City
- **Inputs:**
 - Monthly taxi ride log files
 - The local data set is **small** (~2 MB)
- **Approach:**
 - Preprocess and explore data
 - Develop and validate predictive model (linear fit)
 - Work with this subset of data for prototyping



Big Data capabilities in MATLAB: Integration with SPARK



Analyze Big Data on Spark enabled Hadoop cluster



Program using “tall arrays”

```
tt = tall(ds)
a = tt.ArrDelay;

m = mean(a,'omitnan');
s = std(a,'omitnan');
one_sigma_bounds = [m-s m m+s];
sig1 = gather(one_sigma_bounds)
```

Program using MATLAB API for Spark

```
maprdd = rdd.flatMap(@carrierToCount);
redrdd = maprdd.reduceByKey( @(acc,val
countdata = redrdd.collect());

% Count and display carrier occurrence
count = 0;
for i=1:numel(countdata)
    count = count + out{i}{2};
```

Example: Working with Big Data on a Spark enabled Hadoop Cluster

- **Objective:** Create a model to predict the cost of a taxi ride in New York City
- **Inputs:**
 - Monthly taxi ride log files
 - The full data set is **big** (~11 GB)
- **Approach:**
 - Preprocess and explore data
 - Develop and validate predictive model (linear fit)
 - Use the algorithm prototype developed with Tall Arrays to scale to the full data set on HDFS



Example: Running on Spark + Hadoop

The screenshot displays the MATLAB IDE interface with the following components:

- Current Folder:** Lists files such as `MaxAmountByPaymentType.m`, `TallArrayDemo.m`, and `TallArrayNTCTrainingHortonworks.m`.
- Editor:** Shows the script `TallArrayNTCTrainingHortonworks.m` with the following code:


```

16
17
18 %% Create datastore and select variables of interest
19 % Access data
20 fileLoc = 'hdfs://10.0.4.207:8020/user/matlab/newYorkTaxi/*taxidataNYC_*.csv';
21 ds=datastore(fileLoc)
22 preview(ds)
23
24 %% Identify data of interest and customize options.
25 ds.VariableNames(2:3) = {'pickuptime','dropofftime'};
26 ds.SelectedVariableNames = {'pickuptime','dropofftime','trip_distance',...
27 'payment_type','fare_amount'};
28 ds.SelectedFormats(1:2) = {'%{yyyy-MM-dd HH:mm:ss}D'};
29 preview(ds)
30
31 %% Create tall array backed by datastore
32 tt = tall(ds);
33
34 %% Preprocess data
35 %Determine trip duration
36 tt.hr_of_day = hour(tt.pickuptime);
37 tt.trip_minutes = minutes(tt.dropofftime - tt.pickuptime)
38
39 newCats = {'Credit card', 'Cash', 'No charge', 'Dispute', 'Unknown', 'Voided'};
40 tt.payment_type = categorical(tt.payment_type,1:6,newCats);
41
42 %% Remove outliers with logical indexing.
43 idx = tt.trip_minutes <= 1 | ...
44       tt.trip_minutes >= 90 | ...
45       tt.trip_distance <= 1 | ...
46       tt.trip_distance >= 30 | ...
47       tt.fare_amount <= 0 | ...
48       tt.fare_amount > 100;

```
- Workspace:** Shows a variable `cluster` with a value of `1x1 Hadoop`.
- Command Window:** Shows the prompt `/;> Trial>>`.
- Details:** A section at the bottom left with the text "Select a file to view details".
- Status Bar:** Shows "script", "Ln 6", "Col 10", and "Trial Days Remaining: 83".

Accès à MATLAB

- Accès au réseau du laboratoire
 - Nombre de jetons illimité
 - Accès à toutes les toolboxes de la licence campus
- Accès sur PC personnel



Uniquement aux étudiants et au personnel de l'Université d'Orléans
prenom.nom@univ-orleans.fr
prenom.nom@etu.univ-orleans.fr

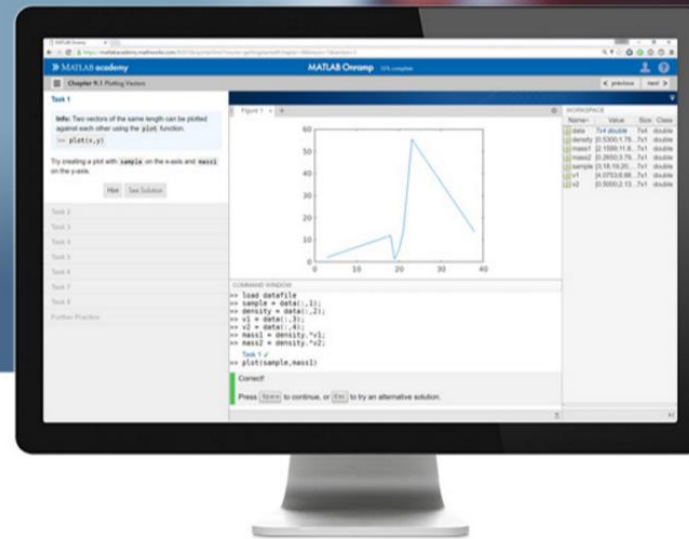
Self-paced MATLAB Tutorials

[Contact Us](#) [How To Buy](#) [Log In](#)

[Products](#) [Solutions](#) [Academia](#) [Support](#) [Community](#) [Events](#) [Company](#)

MATLAB Academy

Search MathWorks.com



Learn MATLAB for Free

[Launch MATLAB Onramp now](#)

<http://matlabacademy.mathworks.com>

Search: MATLAB Academy

Formations en ligne

Pour tous

2- 4 hours content

Uniquement aux étudiants et au personnel de l'Université d'Orléans
prenom.nom@univ-orleans.fr
prenom.nom@etu.univ-orleans.fr



Get Started

MATLAB Onramp

Deep Learning Onramp

Computational Mathematics

*Available only to users at universities that offer campus-wide online training access.

Solving Nonlinear Equations with MATLAB

Solving Ordinary Differential Equations with MATLAB

Introduction to Linear Algebra with MATLAB

Introduction to Statistical Methods with MATLAB

3-7 hours of solving Computational Mathematics with MATLAB

Core MATLAB Functionality

MATLAB Fundamentals

MATLAB Programming Techniques

MATLAB for Financial Applications

Data Analytics

MATLAB for Data Processing and Visualization

Machine Learning with MATLAB

>80 hours of learning how to improve MATLAB skills

Objectifs: Présentation des outils MathWorks pour la pédagogie mis en place pour

Enseignant

- Economiser du temps pour les enseignants



Convergent Series
Leibniz series approximation of π

Problem Description
One of the methods to estimate the value of π is to use the Leibniz series expansion to a reasonably large number of terms and use the expression below to estimate the value of π .

$$\frac{\pi}{4} \approx 1 - \frac{1}{3} + \frac{1}{5} - \frac{1}{7} + \dots + \frac{(-1)^{n+1}}{2n-1} = \sum_{k=1}^n \frac{(-1)^{k+1}}{2k-1}$$

Using this expression, write a script to compute the value of π . Estimate the value of π using 10, 20, ..., 200 terms. Create a plot of the estimated values of π vs. the number of terms used for the estimation. Observe the convergence to the estimated value of π as the number of terms increase.

Use the code template provided to develop your solution.

Solution

```

1 N = 10:10:100;
2 approxP1 = % Initialize a zero vector of appropriate length.
3 for ind = 1:length(N)
4     % Enter your code here.
5
6     approxP1(ind) = % Estimated pi value corresponding to N terms
7
8 end
9
10 estP1 = approxP1(end);
11
12
    
```

Buttons: Reset, Test, Submit

Etudiant

- Motiver et accélérer l'apprentissage des étudiants

Formations en ligne

MATLAB mobile

Live Script editor

Cody Coursework

Connectivité Hardware



Learn MATLAB for Free



Air Mass and Solar Radiation

As light from the sun passes through the earth's atmosphere, some of the solar radiation will be absorbed. The air mass is a function of solar elevation (α). As shown in the diagram below, it is a measure of the amount of air that light travels through the atmosphere (Y) relative to the shortest possible path (X) through the atmosphere.

The larger the air mass, the less radiation reaches the ground. The air mass can be calculated from the equation:

$$AM = \frac{1}{\cos(\alpha)}$$

Then the solar radiation (K_{ext}) reaching the ground can be calculated from the empirical equation:

$$K_{ext} = 1.353 \cdot (0.75)^{AM}$$

AM = 1 / (cos(99 - 41*alpha)) = 0.36572 + 0.07995*alpha^-1 + 6.8241*alpha^-2
 K_ext = 1.353 * (0.75)^(AM) = 0.7191
 K_ext = 1.353 * (0.75)^(AM) = 0.7191
 K_ext = 1.353 * (0.75)^(AM) = 0.7191

Solar Radiation on Fixed Panels

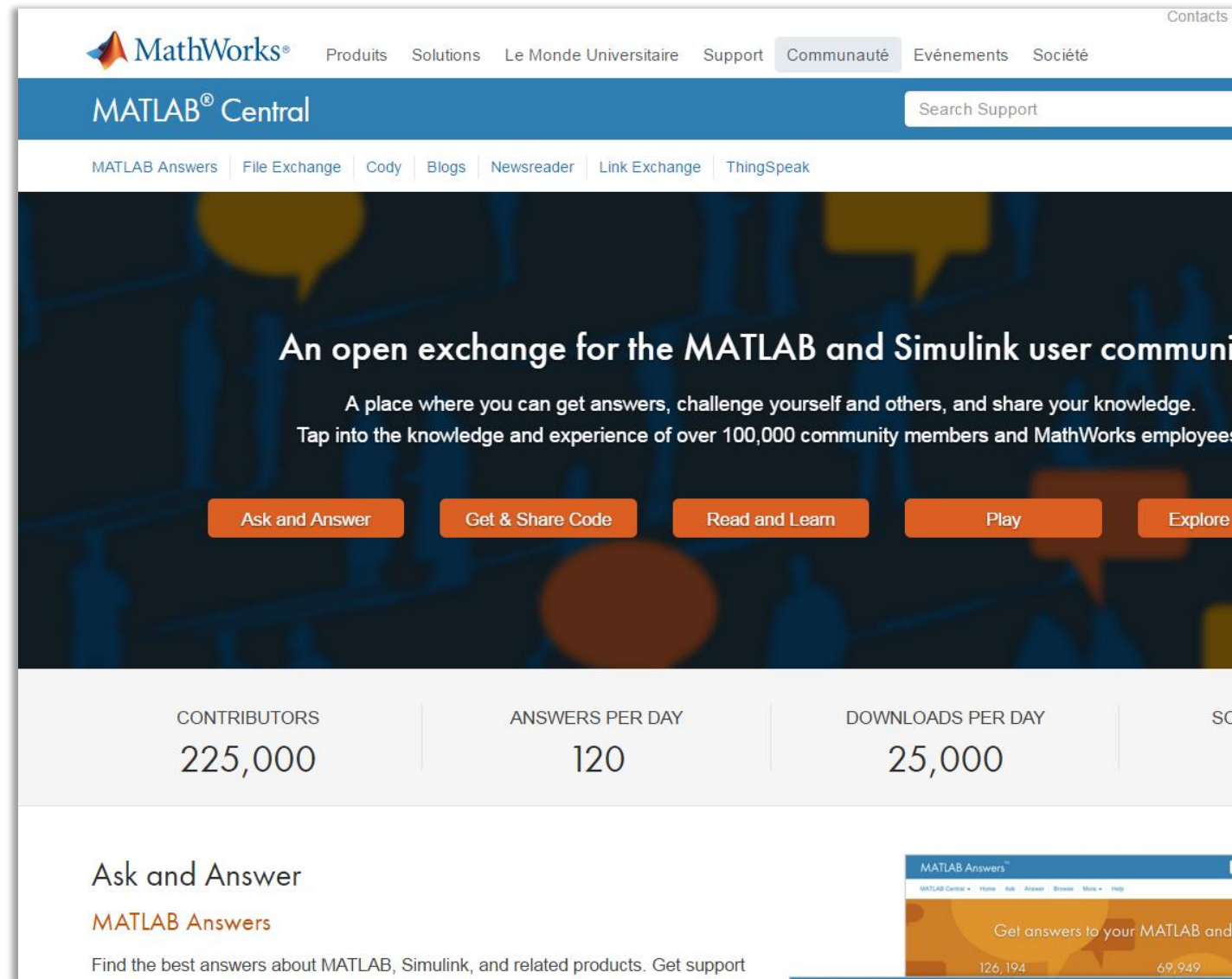
Panels installed with a solar tracker can move with the sun and receive 100% of the sun's radiation as the sun moves across the sky. However, most solar cell installations have panels set at a fixed azimuth and tilt. Therefore the actual radiation reaching the panel will also depend on the sun's position.

Summary

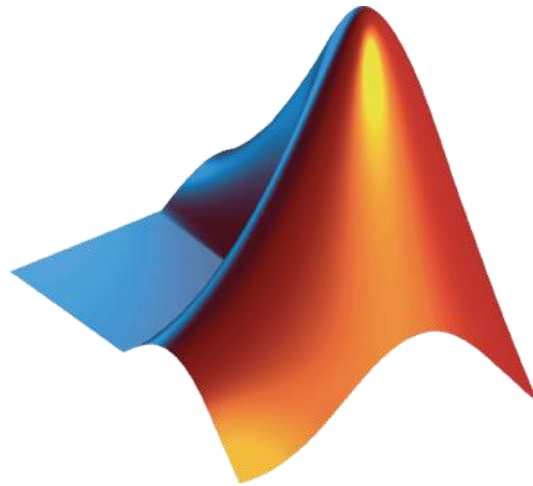
- Easily develop parallel MATLAB applications without being a parallel programming expert
- Speed up the execution of your MATLAB applications using additional hardware
- Develop parallel applications on your desktop and easily scale to a cluster when needed
- Scale-up your MATLAB applications developed locally to Big Data systems

MATLAB Central

- Community for MATLAB and Simulink users
- File Exchange
 - Access free files including functions, apps, examples, and models
- MATLAB Answers
 - Ask programming questions or search community-answered questions
- Newsgroup
 - Participate in technical discussions
- Blogs
 - Read commentary from engineers who design, build, and support MathWorks products



The screenshot shows the MATLAB Central website interface. At the top, there is a navigation bar with the MathWorks logo and links for Products, Solutions, Le Monde Universitaire, Support, Communauté, Événements, and Société. Below this is a blue header with 'MATLAB® Central' and a search bar labeled 'Search Support'. A secondary navigation bar includes links for MATLAB Answers, File Exchange, Cody, Blogs, Newsreader, Link Exchange, and ThingSpeak. The main content area features a large banner with the text 'An open exchange for the MATLAB and Simulink user community' and a sub-headline: 'A place where you can get answers, challenge yourself and others, and share your knowledge. Tap into the knowledge and experience of over 100,000 community members and MathWorks employees.' Below the banner are five orange buttons: 'Ask and Answer', 'Get & Share Code', 'Read and Learn', 'Play', and 'Explore'. A statistics section displays three metrics: 'CONTRIBUTORS 225,000', 'ANSWERS PER DAY 120', and 'DOWNLOADS PER DAY 25,000'. At the bottom, there is a section titled 'Ask and Answer' with a sub-link for 'MATLAB Answers' and a brief description: 'Find the best answers about MATLAB, Simulink, and related products. Get support'. A small inset window shows a preview of the MATLAB Answers page with a search bar and a button to 'Get answers to your MATLAB and Simulink questions'.



Merci

Questions?